

[illegible][illegible]

```
DDDDDDDD  QQQQQQ  DDDDDDD  IIIIII  SSSSSSS  KK  KK  SSSSSSS
DDDDDDDD  QQQQQQ  DDDDDDD  IIIIII  SSSSSSS  KK  KK  SSSSSSS
DD  DD  QQ  QQ  DD  DD  II  SS  SS  KK  KK  SS
DD  DD  QQ  QQ  DD  DD  II  SS  SS  KK  KK  SS
DD  DD  QQ  QQ  DD  DD  II  SS  SS  KK  KK  SS
DD  DD  QQ  QQ  DD  DD  II  SS  SS  KK  KK  SS
DD  DD  QQ  QQ  DD  DD  II  SS  SS  KK  KK  SS
DD  DD  QQ  QQ  DD  DD  II  SS  SS  KK  KK  SS
DD  DD  QQ  QQ  DD  DD  II  SS  SS  KK  KK  SS
DDDDDDDD  QQQQ  QQ  DDDDDDD  IIIIII  SSSSSSS  SS  SS  SSSSSSS
DDDDDDDD  QQQQ  QQ  DDDDDDD  IIIIII  SSSSSSS  SS  SS  SSSSSSS

LL  IIIIII  SSSSSSS
LL  IIIIII  SSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL  IIIIII  SSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSS
```



```
integer*4      control_status_register
integer*4      bus_address_register
```

```

0273 integer*4 byte_control_register
0274 integer*4 disk_address_register
0275 integer*4 multi_purpose_register
0276 integer*4 ecc_position_register
0277 integer*4 ecc_pattern_register
0278 integer*4 data_path_number
0279 integer*4 data_path_register
0280 integer*4 final_map_register
0281 integer*4 previous_map_register
0282 integer*4 vec$l_mapreg
0283
0284 equivalence (emb$l_dv_regsav(0),control_status_register)
0285 equivalence (emb$l_dv_regsav(1),bus_address_register)
0286 equivalence (emb$l_dv_regsav(2),byte_control_register)
0287 equivalence (emb$l_dv_regsav(3),disk_address_register)
0288 equivalence (emb$l_dv_regsav(4),multi_purpose_register)
0289 equivalence (emb$l_dv_regsav(5),ecc_position_register)
0290 equivalence (emb$l_dv_regsav(6),ecc_pattern_register)
0291 equivalence (emb$l_dv_regsav(7),data_path_number)
0292 equivalence (emb$l_dv_regsav(8),data_path_register)
0293 equivalence (emb$l_dv_regsav(9),final_map_register)
0294 equivalence (emb$l_dv_regsav(10),previous_map_register)
0295 equivalence (emb$l_dv_regsav(11),vec$l_mapreg)
0296
0297 character*12 v1csr(0:0)
0298 data v1csr(0) /*DRIVE READY*/
0299
0300 character*17 v2csr(6:7)
0301 data v2csr(6) /*INTERRUPT ENABLE*/
0302 data v2csr(7) /*CONTROLLER READY*/
0303
0304 character*21 v3csr(10:10)
0305 data v3csr(10) /*OPERATION INCOMPLETE*/
0306
0307 character*20 v4csr(13:15)
0308 data v4csr(13) /*NON-EXISTENT MEMORY*/
0309 data v4csr(14) /*DRIVE ERROR*/
0310 data v4csr(15) /*COMPOSITE ERROR*/
0311
0312 character*22 v5csr(22:24)
0313 data v5csr(22) /*R80 SKIP SECTOR ERROR*/
0314 data v5csr(23) /*R80 SKIP SECTOR ERROR*/
0315 data v5csr(24) /*INTERRUPT REQUEST*/
0316
0317 character*30 v6csr(26:28)
0318 data v6csr(26) /*R80*/
0319 data v6csr(27) /*AUTOMATIC SKIP SECTOR INHIBIT*/
0320 data v6csr(28) /*TIMEOUT INHIBIT*/
0321
0322 character*11 v1rl02_mpr(3:5)
0323 data v1rl02_mpr(3) /*BRUSH HOME*/
0324 data v1rl02_mpr(4) /*HEADS OUT*/
0325 data v1rl02_mpr(5) /*COVER OPEN*/
0326
0327 character*19 v2rl02_mpr(8:15)
0328 data v2rl02_mpr(8) /*DRIVE SELECT ERROR*/
0329 data v2rl02_mpr(9) /*VOLUME CHECK*/

```



```

0330      data      v2rl02_mpr(10)  /*WRITE GATE ERROR*/
0331      data      v2rl02_mpr(11)  /*SPINDLE ERROR*/
0332      data      v2rl02_mpr(12)  /*SEEK TIMEOUT*/
0333      data      v2rl02_mpr(13)  /*WRITE LOCK*/
0334      data      v2rl02_mpr(14)  /*HEAD CURRENT ERROR*/
0335      data      v2rl02_mpr(15)  /*WRITE DATE ERROR*/
0336
0337      character*14 v1r80_mpr(8:13)
0338      data      v1r80_mpr(8)    /*FAULT*/
0339      data      v1r80_mpr(9)    /*PLUG VALID*/
0340      data      v1r80_mpr(10)   /*SEEK ERROR*/
0341      data      v1r80_mpr(11)   /*ON CYLINDER*/
0342      data      v1r80_mpr(12)   /*DRIVE READY*/
0343      data      v1r80_mpr(13)   /*WRITE PROTECT*/
0344
0345      integer*4    compress4
0346      integer*4    compressc
0347      integer*4    field
0348
0349      character*27 idc_command(0:7)
0350      data      idc_command(0)  /*NO DRIVE OPERATION*/
0351      data      idc_command(1)  /*WRITE CHECK DATA*/
0352      data      idc_command(2)  /*GET STATUS*/
0353      data      idc_command(3)  /*SEEK*/
0354      data      idc_command(4)  /*READ HEADER*/
0355      data      idc_command(5)  /*WRITE DATA*/
0356      data      idc_command(6)  /*READ DATA*/
0357      data      idc_command(7)  /*READ DATA W/O HEADER CHECK*/
0358
0359      logical*1     diagnostic_mode
0360
0361      integer*4      lib$extzv
0362      integer*4      data_check_and_opi_bits
0363      integer*4      data_late_and_opi_bits
0364      integer*4      sector_count
0365      integer*4      ecc_status_bits
0366      integer*4      rl02_status_bits
0367
0368      character*20   v1rl02_status_bits(0:7)
0369      data      v1rl02_status_bits(0) /*LOAD STATE*/
0370      data      v1rl02_status_bits(1) /*SPIN UP*/
0371      data      v1rl02_status_bits(2) /*BRUSH CYCLE*/
0372      data      v1rl02_status_bits(3) /*LOAD HEADS*/
0373      data      v1rl02_status_bits(4) /*SEEK TRACK COUNTING*/
0374      data      v1rl02_status_bits(5) /*SEEK LINEAR MODE*/
0375      data      v1rl02_status_bits(6) /*UNLOAD HEADS*/
0376      data      v1rl02_status_bits(7) /*SPIN DOWN*/
0377
0378      integer*4      device_function
0379      integer*4      device_type
0380      integer*4      sector
0381      integer*4      cylinder
0382      integer*4      tag
0383      integer*4      head
0384
0385      character*11   vldar(0:1)
0386      data      vldar(0) /*MARKER*/

```

```
0387      data          v1dar(1)      /*GET STATUS*/
0388
0389      character*6     v2dar(3:3)
0390      data            v2dar(3)      /*RESET*/
0391
0392      character*8     v4dar(2:2,0:1)
0393      data            v4dar(2,0)    /*REVERSE*/
0394      data            v4dar(2,1)    /*FORWARD*/
0395
0396      character*18    v6dar(4:4,0:1)
0397      data            v6dar(4,0)    /*SELECT LOWER HEAD*/
0398      data            v6dar(4,1)    /*SELECT UPPER HEAD*/
0399
0400      character*15    v7dar(6:6)
0401      data            v7dar(6)      /*RETURN-TO-ZERO*/
0402
0403
0404      call frctof (lun)
0405
0406      call dhead1 (lun,'RB730')
0407
0408      diagnostic_mode = .false.
0409
0410      if (lib$extzv(25,1,control_status_register) .eq. 1)
0411      1 diagnostic_mode = .true.
0412
0413      device_function = lib$extzv (1,3,control_status_register)
0414
0415      device_type = lib$extzv (26,1,control_status_register)
0416
0417      call linchk (lun,2)
0418
0419      write(lun,5) 'RB CSR',control_status_register
0420      format(/' ',t8,a,t24,z8.8)
0421
0422      if (.not. diagnostic_mode) then
0423
0424      call output (lun,control_status_register,v1csr,0,0,0,'0')
0425
0426      call linchk (lun,1)
0427
0428      if (lib$extzv(29,1,control_status_register) .eq. 1) then
0429
0430      10 write(lun,10) 'R80 WRITE FORMAT FUNCTION'
0431      format(' ',t40,a)
0432      else
0433
0434      idc_function = lib$extzv(1,3,control_status_register)
0435
0436      15 write(lun,15) idc_command(idc_function)
0437      format(' ',t40,a,compressc (idc_command(idc_function)))
0438      endif
0439
0440      call output (lun,control_status_register,v2csr,6,6,7,'0')
0441
0442      call linchk (lun,1)
0443
```

```
0444 write(lun,20) 'DRIVE #',lib$extzv(8,2,control_status_register),
0445 1 ' . SELECTED'
0446 20 format(' ',t40,a,i1.1,a)
0447
0448 call output (lun,control_status_register,v3csr,10,10,10,'0')
0449
0450 data_check_and_opi_bits = lib$extzv(10,2,control_status_register)
0451
0452 if (
0453 1 data_check_and_opi_bits .eq. 2
0454 1 .or.
0455 1 data_check_and_opi_bits .eq. 3
0456 1 ) then
0457
0458 call linchk (lun,1)
0459 endif
0460
0461 if (data_check_and_opi_bits .eq. 2) then
0462
0463 25 write(lun,25) 'DATA CHECK ERROR'
0464 format(' ',t40,a)
0465
0466 else if (data_check_and_opi_bits .eq. 3) then
0467
0468 write(lun,25) 'HEADER CRC ERROR'
0469 endif
0470
0471 data_late_and_opi_bits = lib$extzv(10,3,control_status_register)
0472
0473 if (
0474 1 data_late_and_opi_bits .eq. 4
0475 1 .or.
0476 1 data_late_and_opi_bits .eq. 5
0477 1 ) then
0478
0479 call linchk (lun,1)
0480 endif
0481
0482 if (data_late_and_opi_bits .eq. 4) then
0483
0484 write(lun,25) 'DATA LATE'
0485
0486 else if (data_late_and_opi_bits .eq. 5) then
0487
0488 write(lun,25) 'HEADER NOT FOUND'
0489 endif
0490
0491 call output (lun,control_status_register,v4csr,13,13,15,'0')
0492
0493 do 35,i = 16,19
0494
0495 if (lib$extzv(i,1,control_status_register) .eq. 1) then
0496
0497 call linchk (lun,1)
0498
0499 30 write(lun,30) 'ATTENTION DRIVE #',i-16,'.'
0500 format(' ',t40,a,i1.1,a)
```


000
000
000
000
000
000
000
000
000
001
001
001
001
001
001
001
001
001
002
002
002
002
002
002
002
002
002
002
003
003
003
003
003
003
003
003
003
003
004
004
004
004
004
004
004
004
004
005
005
005
005
005
005
005

46

[illegible]

```

0615      call output (lun,disk_address_register,v1dar,0,0,1,'0')
0616
0617      call output (lun,disk_address_register,v4dar,2,2,2,'2')
0618
0619      call output (lun,disk_address_register,v2dar,3,3,3,'0')
0620
0621      call output (lun,disk_address_register,v6dar,4,4,4,'2')
0622
0623      cylinder = lib$extzv (7,9,disk_address_register)
0624
0625      call linchk (lun,1)
0626
0627      write(lun,47) cylinder
47      format(' ',t40,i<compress4 (cylinder)>,'. CYLINDER(S) TO MOVE')
0629
0630      else if (device_type .eq. 1) then
0631
0632      tag = lib$extzv (13,3,disk_address_register)
0633
0634      call linchk (lun,1)
0635
0636      if (tag .eq. 1) then
0637
0638      cylinder = lib$extzv (0,10,disk_address_register)
0639
0640      write(lun,48) 'CYLINDER #',cylinder
48      format(' ',t40,a,i<compress4 (cylinder)>,'. SELECTED')
0642
0643      else if (tag .eq. 2) then
0644
0645      head = lib$extzv (0,4,disk_address_register)
0646
0647      write(lun,48) 'HEAD #',head
0648
0649      else if (tag .eq. 4) then
0650
0651      call output (lun,disk_address_register,v7dar,6,6,6,'0')
0652      endif
0653      endif
0654      endif
0655      endif
0656
0657      call linchk (lun,1)
0658
0659      write(lun,50) 'RB MPR',multi_purpose_register
50      format(' ',t8,a,t24,z8.8)
0661
0662      if (.not. diagnostic_mode) then
0663
0664      if (lib$extzv (26,1,control_status_register) .eq. 1) then
0665
0666      sector_count = lib$extzv (0,5,multi_purpose_register)
0667
0668      call linchk (lun,1)
0669
0670      write(lun,55) 'SECTOR COUNT ',sector_count,'.'
55      format(' ',t40,a,i<compress4 (sector_count)>,'a)
0671

```


[illegible]

038
038
038
039
039
039
039
039
039
039
039
039
040
040
040
040
040
040
040
040
040
041
041
041
041
041
041
041
041
041
042
042
042
042
042
042
042
042
042
042
043
043
043
043
043
043
043
043

PROGRAM SECTIONS

Name	Bytes	Attributes							
0 \$CODE	2861	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT LONG
1 \$PDATA	634	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT LONG
2 \$LOCAL	2972	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT LONG
3 EMB	512	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT LONG
Total Space Allocated		6979							

ENTRY POINTS

Address	Type	Name
0-00000000		DQDISKS

VARIABLES

Address	Type	Name	Address	Type	Name
3-00000056	I*4	BUS ADDRESS REGISTER	3-0000005A	I*4	BYTE CONTROL_REGISTER
3-00000052	I*4	CONTROL STATUS_REGISTER	2-00000424	I*4	CYLINDER
2-00000404	I*4	DATA_CHECK_AND_OPI_BITS	2-00000408	I*4	DATA_LATE_AND_OPI_BITS
3-0000006E	I*4	DATA_PATH_NUMBER	3-00000072	I*4	DATA_PATH_REGISTER
2-00000418	I*4	DEVICE FUNCTION	2-0000041C	I*4	DEVICE TYPE
2-000003FF	L*1	DIAGNOSTIC MODE	3-0000005E	I*4	DISK ADDRESS_REGISTER
3-0000006A	I*4	ECC_PATTERN_REGISTER	3-00000066	I*4	ECC_POSITION_REGISTER
2-00000410	I*4	ECC_STATUS_BITS	3-0000001C	L*1	EMBSB_DV_CLASS
3-00000010	L*1	EMBSB_DV_ERTCNT	3-00000011	L*1	EMBSB_DV_ERTMAX
3-0000003E	L*1	EMBSB_DV_NAMLNG	3-0000003A	L*1	EMBSB_DV_SLAVE
3-0000001D	L*1	EMBSB_DV_TYPE	3-00000036	I*4	EMBSL_DV_CHAR
3-00000012	I*4	EMBSL_DV_IOSB1	3-00000016	I*4	EMBSL_DV_IOSB2
3-00000026	I*4	EMBSL_DV_MEDIA	3-0000004E	I*4	EMBSL_DV_NUMREG
3-0000002E	I*4	EMBSL_DV_OPCNT	3-00000032	I*4	EMBSL_DV_OWNUIC
3-0000001E	I*4	EMBSL_DV_RQPID	3-00000000	I*4	EMBSL_HD_SID
3-0000003F	CHAR	EMBST_DV_NAME	3-00000024	I*2	EMBSW_DV_BCNT
3-00000022	I*2	EMBSW_DV_BOFF	3-0000002C	I*2	EMBSW_DV_ERRCNT
3-0000003C	I*2	EMBSW_DV_FUNC	3-0000001A	I*2	EMBSW_DV_STS
3-0000002A	I*2	EMBSW_DV_UNIT	3-00000004	I*2	EMBSW_HD_ENTRY
3-0000000E	I*2	EMBSW_HD_ERRSEQ	2-00000400	I*4	FIELD
3-00000076	I*4	FINAL_MAP_REGISTER	2-0000042C	I*4	HEAD
2-00000434	I*4	I	2-00000430	I*4	IDC FUNCTION
AP-00000004	L*1	LUN	3-00000062	I*4	MULTI PURPOSE_REGISTER
3-0000007A	I*4	PREVIOUS_MAP_REGISTER	2-00000414	I*4	RL02 STATUS_BITS
2-00000420	I*4	SECTOR	2-0000040C	I*4	SECTOR COUNT
2-00000428	I*4	TAG	3-0000007E	I*4	VECSL_MAPREG

DR1
PRO
0
1
2
3
ENT
0
VAR
2
3
4
5
6
7
8
9
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
ARR
M

Subroutine DQDISKS_QIO (lun,emb\$w_dv_func)

include 'src\$:qiocommon.for /nolist'

```

byte          lun
integer*2     emb$w_dv_func
integer*4     qiocode(0:1,0:63)

if (qiocode(0,0) .eq. 0) then
qiocode(1,00) = %loc(io$_nop)
qiocode(1,02) = %loc(io$_seek)
qiocode(1,03) = %loc(io$_recal)
qiocode(1,04) = %loc(io$_drvclr)
qiocode(1,08) = %loc(io$_packack)
qiocode(1,10) = %loc(io$_writecheck)
qiocode(1,11) = %loc(io$_writepblk)
qiocode(1,12) = %loc(io$_readpblk)
qiocode(1,14) = %loc(io$_readhead)
qiocode(1,26) = %loc(io$_setchar)
qiocode(1,27) = %loc(io$_sensechar)
qiocode(1,32) = %loc(io$_writelblk)
qiocode(1,33) = %loc(io$_readlblk)
qiocode(1,35) = %loc(io$_setmode)
qiocode(1,39) = %loc(io$_sensemode)
qiocode(1,48) = %loc(io$_writevblk)

```

```
0321      qiocode(1,49) = %loc(io$_readvblk)
0322
0323      qiocode(1,50) = %loc(io$_access)
0324
0325      qiocode(1,51) = %loc(io$_create)
0326
0327      qiocode(1,52) = %loc(io$_deaccess)
0328
0329      qiocode(1,53) = %loc(io$_delete)
0330
0331      qiocode(1,54) = %loc(io$_modify)
0332
0333      qiocode(1,56) = %loc(io$_acpcontrol)
0334
0335      qiocode(1,57) = %loc(io$_mount)
0336
0337      do 10,i = 0,63
0338
0339      qiocode(0,i) = 33
0340
0341      if (qiocode(1,i) .eq. 0) then
0342
0343      qiocode(1,i) = %loc(qio_string)
0344      endif
0345
0346      10 continue
0347      endif
0348
0349      call irp$w_func (lun,emb$w_dv_func,
0350      1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))
0351
0352      return
0353
0354      end
```


PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	255	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	8	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	548	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 QIOCOMMON	1247	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	2058	

ENTRY POINTS

Address	Type	Name
0-00000000		DQDISKS_QIO

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008a	I*2	EMBSW DV FUNC	2-00000200	I*4	I
3-00000442	CHAR	IOS_ABORT	3-00000340	CHAR	IOS_ACCESS
3-000003C2	CHAR	IOS_ACPCONTROL	3-000004B3	CHAR	IOS_AVAILABLE
3-00000297	CHAR	IOS_CLEAN	3-00000369	CHAR	IOS_CREATE
3-00000385	CHAR	IOS_DEACCESS	3-00000393	CHAR	IOS_DELETE
3-0000026D	CHAR	IOS_DIAGNOSE	3-00000065	CHAR	IOS_DRVCLR
3-000004CB	CHAR	IOS_DSE	3-000000A9	CHAR	IOS_ERASETAPE
3-00000276	CHAR	IOS_FORMAT	3-00000071	CHAR	IOS_INITIALIZE
3-00000014	CHAR	IOS_LOADMCODE	3-000003A1	CHAR	IOS_MODIFY
3-000003E2	CHAR	IOS_MOUNT	3-00000000	CHAR	IOS_NOP
3-0000009D	CHAR	IOS_OFFSET	3-000000EB	CHAR	IOS_PACKACK
3-000000E0	CHAR	IOS_QSTOP	3-000003EF	CHAR	IOS_RDSTATS
3-00000421	CHAR	IOS_READCSR	3-00000169	CHAR	IOS_READHEAD
3-000002B6	CHAR	IOS_READLBLK	3-0000013F	CHAR	IOS_READPBLK
3-00000200	CHAR	IOS_READPRESET	3-00000195	CHAR	IOS_READTRACKD
3-0000033A	CHAR	IOS_READVBLK	3-0000045A	CHAR	IOS_READWTHBUF
3-00000484	CHAR	IOS_READWTHXBUF	3-0000004D	CHAR	IOS_RECAL
3-0000007C	CHAR	IOS_RELEASE	3-000001AB	CHAR	IOS_REREADN
3-000001B8	CHAR	IOS_REREADP	3-000000CA	CHAR	IOS_RETCENTER
3-000002E6	CHAR	IOS_REWIND	3-000002C9	CHAR	IOS_REWINDOFF
3-000000FC	CHAR	IOS_SEARCH	3-00000024	CHAR	IOS_SEEK
3-00000231	CHAR	IOS_SENSECHAR	3-00000309	CHAR	IOS_SENSEMODE
3-0000021D	CHAR	IOS_SETCHAR	3-000003B8	CHAR	IOS_SETCLOCK
3-00000088	CHAR	IOS_SETCLOCKP	3-000002DD	CHAR	IOS_SETMODE
3-000002ED	CHAR	IOS_SKIPFILE	3-000002FA	CHAR	IOS_SKIPRECORD
3-00000029	CHAR	IOS_SPACEFILE	3-0000010E	CHAR	IOS_SPACERECORD
3-000003D7	CHAR	IOS_STARTDATA	3-000000B4	CHAR	IOS_STARTDATAP
3-00000037	CHAR	IOS_STARTMPROC	3-0000020F	CHAR	IOS_STARTSPNDL
3-00000059	CHAR	IOS_STOP	3-0000000D	CHAR	IOS_UNLOAD
3-00000468	CHAR	IOS_WRITEBUFNCRC	3-0000011E	CHAR	IOS_WRITECHECK
3-000001E4	CHAR	IOS_WRITECHECKH	3-000003FF	CHAR	IOS_WRITECSR
3-00000153	CHAR	IOS_WRITEHEAD	3-000002A2	CHAR	IOS_WRITELBLK
3-00000247	CHAR	IOS_WRITEMARK	3-00000314	CHAR	IOS_WRITEOF
3-0000012A	CHAR	IOS_WRITEPBLK	3-000001C9	CHAR	IOS_WRITERET

DQDISKS_QIO

D 8
16-Sep-1984 00:02:07
5-Sep-1984 13:52:37

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DQDISKS.FOR;1 Page 17

3-0000017E CHAR IOS_WRIETRACKD
3-00000448 CHAR IOS_WRIETWTHBUF
AP-00000004a L*1 LUN

3-00000326 CHAR IOS_WRITEVBLK
3-00000257 CHAR IOS_WRTTMKR
3-000004A1 CHAR QIO_STRING

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	I*4	QIOCODE	512	(0:1, 0:63)

LABELS

Address	Label
**	10

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	IRPSW_FUNC	I*4	LIB\$EXTZV

E 8
16-Sep-1984 00:02:07
5-Sep-1984 13:52:37

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DQDISKS.FOR;1 Page 18

0001
0002

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:DQDISKS/OBJ=OBJ\$:DQDISKS MSRC\$:DQDISKS

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/STANDARD=(NOSYNTAX,NOSOURCE FORM)

/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)

/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 10.94 seconds

Elapsed Time: 25.02 seconds

Page Faults: 269

Dynamic Memory: 248 pages

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938

0147

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY